

## گزارش تمرین دوم

آشنایی با پروتکل‌های ارتباطی انتقال داده و سنسورها

سیستم‌های نهفته بی‌درنگ

اعضای گروه:

مبینا شاه‌بنده، غزل کلهر، امید بدایق، دیار محمدی

بهار ۱۴۰۰

## فهرست مطالب

3	توضیحات کد
3	Main-Board
9	TH-Board
12	شبیه سازی در پروتئوس
16	پاسخ سوال 1
16	فرکانس مورد استفاده برای ارتباط بی سیم در بلوتوث
17	پاسخ سوال 2
19	پاسخ سوال 3

## توضیحات کد

### Main-Board

در ابتدا کتابخانه‌های مورد استفاده (شامل کتابخانه Arduino و LiquidCrystal) را include می‌کنیم. سپس جداکننده‌های داده‌ی ارسالی برای دما و رطوبت را تعیین کرده و سپس پین‌های ورودی موتور DC و حداکثر سرعت آن را تعریف می‌کنیم.

```
#include <Arduino.h>

#include <LiquidCrystal.h>

#define TEMPERATURE_DELIMITER '*'

#define HUMIDITY_DELIMITER '$'

#define DC_MOTOR_PIN1 12

#define DC_MOTOR_PIN2 11

#define MOTOR_MAX_SPEED 255

int motorSpeed;

int cnt;
```

سپس LCD را تعریف کرده و پین‌های آن را هم تعیین می‌کنیم.

```
const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal LCD(rs, en, d4, d5, d6, d7);

float humidity, temperature;

int wateringRate;

bool readNewData;

void setWateringRate();

void readSerial();

void showOnLCD();
```

حال تابع `setup` که در ابتدای شبیه سازی اجرا می شود را به صورتی می نویسیم که در آن سرعت موتور 0 باشد (زیرا در ابتدای کار آبیاری صورت نمی گیرد) و پین های موتور را هم به صورت `OUTPUT` تعیین می کنیم (خروجی `Arduino` ورودی دو سر موتور می شود). متغیر `readNewData` به این منظور استفاده شده است که هنگامی که داده ی جدیدی از بلوتوث دریافت می شود، تصمیمات آبیاری بر اساس آن اخذ شود. نرخ آبیاری نیز در ابتدا 0 است. ابعاد `LCD` و `baud rate` ترمینال را نیز تعیین می کنیم که 20 در 4 ابعاد مورد نظر ما در این پروژه و نرخ 9600 مقدار پیش فرض `Arduino` است.

```
void setup()
{
    motorSpeed = 0;
    cnt = 0;
    pinMode (DC_MOTOR_PIN1, OUTPUT);
    pinMode (DC_MOTOR_PIN2, OUTPUT);
    readNewData = false;
    wateringRate = 0;
    LCD.begin(20, 4);
    Serial.begin(9600);
}
```

تابع `loop` که همواره در حال اجرا است به صورتی نوشته می شود که در آن موتور `DC` بچرخد (با تنظیم مقدار دو سر آن که در صورت 1 و 0 بودن می چرخد)، داده ی بلوتوث در صورت دریافت خوانده شود، تصمیمات آبیاری اخذ شود و اطلاعات روی `LCD` نمایش داده شود. شرط `Serial.available() > 4` به این معناست که 4 بایت

عدد شناور دریافت شده باشد و علاوه بر آن یک جداکننده نیز در ابتدای آن داریم (برای دما و برای رطوبت). پس اگر بیش از 4 بایت (از 5 بایت و بیشتر) داده از بلوتوث داشته باشیم می‌توانیم عملیات خواندن این داده‌ها را آغاز کنیم که نحوه‌ی آن در ادامه شرح داده شده است.

```
void loop()
{
    cnt = (cnt + 1) % MOTOR_MAX_SPEED;
    if (cnt < motorSpeed) {
        digitalWrite(DC_MOTOR_PIN1, HIGH);
        digitalWrite(DC_MOTOR_PIN2, LOW);
    } else {
        digitalWrite(DC_MOTOR_PIN1, LOW);
        digitalWrite(DC_MOTOR_PIN2, LOW);
    }
    if (Serial.available() > 4) {
        readSerial();
    }
    if (readNewData) {
        setWateringRate();
        showOnLCD();
        setPWM();
        readNewData = !readNewData;
    }
}
```

برای تنظیم سرعت موتور، PWM آن را به صورتی تنظیم می‌کنیم که اگر نرخ آبیاری 20 باشد، PWM برابر با یک چهارم سرعت ماکسیمم موتور شود. به همین شیوه در صورتی که نرخ آبیاری 10 باشد، PWM برابر با یک دهم سرعت ماکسیمم موتور می‌شود. در غیر این صورت آبیاری نباید صورت بگیرد، پس سرعت موتور باید 0 باشد (موتور نچرخد).

برای تنظیم نرخ آبیاری، شروط مورد نظر را به صورتی می‌نویسیم که اگر رطوبت دریافتی از سنسور بیش از 50 باشد آبیاری صورت نگیرد (نرخ آبیاری 0). در صورتی که رطوبت از 20 کمتر باشد، نرخ آبیاری 20 تعیین می‌شود و در غیر این صورت (رطوبت بین 20 تا 50 باشد) اگر دما کمتر از 25 درجه باشد، آبیاری صورت نمی‌گیرد و اگر دما بیشتر یا مساوی 25 درجه باشد نرخ آبیاری 10 تعیین می‌شود.

```
void setWateringRate()
{
    if (humidity > 50) {
        wateringRate = 0;
        motorSpeed = 0;
    } else if (humidity < 20) {
        wateringRate = 20;
        motorSpeed = MOTOR_MAX_SPEED / 4;
    } else {
        if (temperature < 25) {
            wateringRate = 0;
            motorSpeed = 0;
        }
        if (temperature >= 25) {
            wateringRate = 10;
            motorSpeed = MOTOR_MAX_SPEED / 10;
        }
    }
}
```

```

    }
}
}

```

برای خواندن از بلوتوث، ابتدا کاراکتر آغازین خوانده می‌شود. در صورتی که این کاراکتر جداکننده دما باشد، 4 بایت دیگر به عنوان یک عدد شناور خوانده می‌شود و دما برابر با این عدد قرار داده می‌شود. همچنین در صورتی که دما تغییر کرده باشد باید تصمیم‌گیری نرخ آبیاری صورت گیرد؛ پس متغیر `readNewData` که در ابتدا نیز به آن اشاره شده بود، برابر با `true` می‌شود. در صورتی که کاراکتر آغازین کاراکتر جداکننده رطوبت باشد نیز به همین صورت برای رطوبت عمل می‌کنیم.

```

void readSerial()
{
    char curr = Serial.read();

    if (curr == TEMPERATURE_DELIMITER) {
        float temp = temperature;
        temperature = Serial.parseFloat();
        if (temperature != temp)
            readNewData = true;
    }

    else if (curr == HUMIDITY_DELIMITER) {
        float temp = humidity;
        humidity = Serial.parseFloat();
        if (humidity != temp)
            readNewData = true;
    }
}

```

```
else
    Serial.read();
}
```

در انتها برای نمایش روی LCD ابتدا cursor آن را در گوشه تصویر قرار داده و مقادیر دما و رطوبت و نرخ آبیاری را به ترتیب نمایش می‌دهیم.

```
void showOnLCD()
{
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Temperature: ");
    LCD.println(String(temperature).c_str());
    LCD.setCursor(0, 1);
    LCD.print("Humidity: ");
    LCD.println(String(humidity).c_str());
    LCD.setCursor(0, 2);
    if (wateringRate == 0) {
        LCD.println("No need to water.");
    } else {
        LCD.print("Watering rate: ");
        LCD.print(String(wateringRate) + " ");
        LCD.println("CC");
    }
}
```



## TH-Board

در ابتدا کتابخانه‌های مورد نیاز (شامل کتابخانه Arduino و کتابخانه Wire) را include می‌کنیم. سپس جداکننده‌های دما و رطوبت و آدرس SHT25 I2C را تعیین می‌کنیم (این آدرس از سایت Arduino بدست آمده است).

```
#include <Arduino.h>

#include <Wire.h>

#define TEMPERATURE_DELIMITER '*'
#define HUMIDITY_DELIMITER '$'

#define Addr 0x40

float temperature, humidity;

void getTemperatureFromSensor();

void getHumidityFromSensor();
```

در تابع setup تنها Wire را برای برقراری ارتباط I2C آغاز می‌کنیم و سریال را هم برای ارتباط میان TH-Board و Main Board با baud rate پیش فرض 9600 آغاز می‌کنیم. در تابع loop همواره دما و رطوبت را از سنسور دریافت می‌کنیم و آن را در سریال می‌نویسیم تا Main Board آن را دریافت کند. برای نوشتن دما و رطوبت، ابتدا جداکننده هر کدام را می‌نویسیم و سپس مقدار آن‌ها را.

```
void setup()

{

    Wire.begin();

    Serial.print(TEMPERATURE_DELIMITER);

    Serial.println(temperature);
```

```

Serial.print(HUMIDITY_DELIMITER);

Serial.println(humidity);

}

void loop()
{

  getTemperatureFromSensor();

  getHumidityFromSensor();

}

```

در انتها برای دریافت دما و رطوبت به شکل مشابهی عمل می‌کنیم با این تفاوت که کاراکتری که برای آغاز ارتباط I2C می‌نویسیم در این دو متفاوت است (برای دما کاراکتر `0xF3` و برای رطوبت کاراکتر `0xF5` را می‌نویسیم). همچنین ضربایی که برای بدست آوردن دما و رطوبت در خروجی سنسور ضرب و تقسیم می‌شوند در این دو متفاوت است. این ضرایب از سایت Arduino بدست آمده‌اند.

```

void getTemperatureFromSensor()
{

  unsigned int data[2];

  Wire.beginTransmission(Addr);

  Wire.write(0xF3);

  Wire.endTransmission();

  delay(500);

  Wire.requestFrom(Addr, 2);

  if (Wire.available() == 2) {

    data[0] = Wire.read();

    data[1] = Wire.read();

```

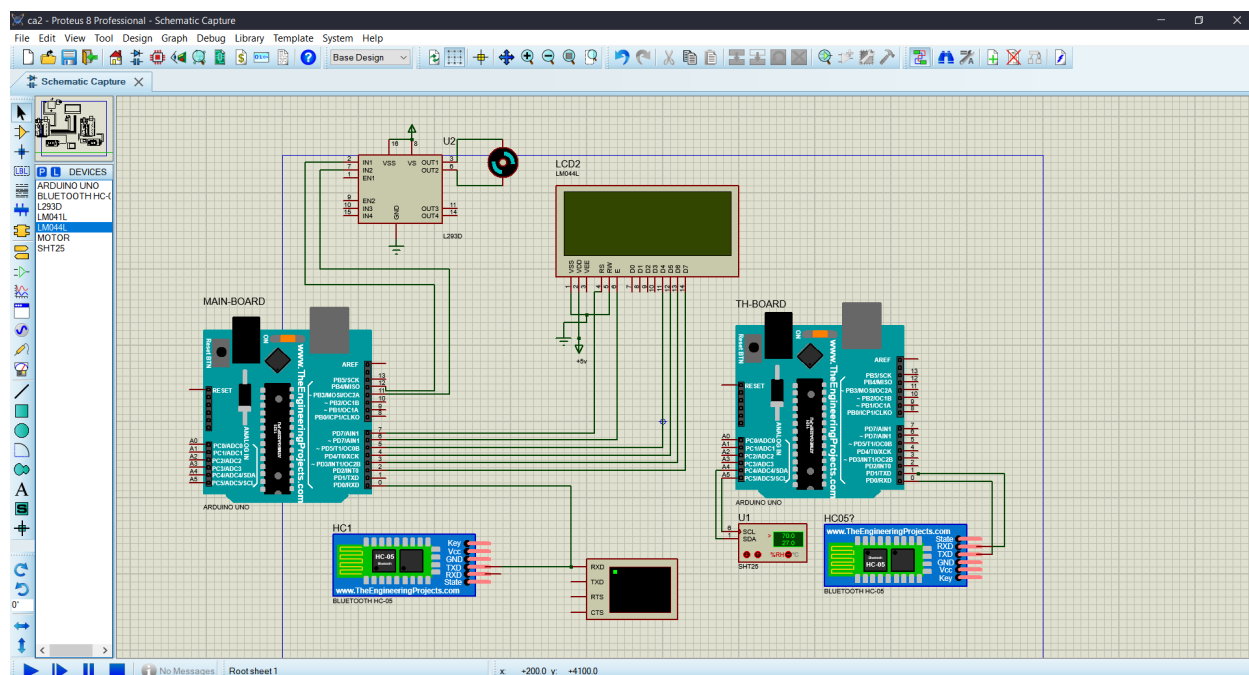
```

    temperature = (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0)
- 46.85;
}
}

void getHumidityFromSensor()
{
    unsigned int data[2];
    Wire.beginTransmission(Addr);
    Wire.write(0xF5);
    Wire.endTransmission();
    delay(500);
    Wire.requestFrom(Addr, 2);
    if (Wire.available() == 2) {
        data[0] = Wire.read();
        data[1] = Wire.read();
        humidity = (((data[0] * 256.0 + data[1]) * 125.0) / 65536.0) - 6;
    }
}

```

## شبیه سازی در پروتئوس



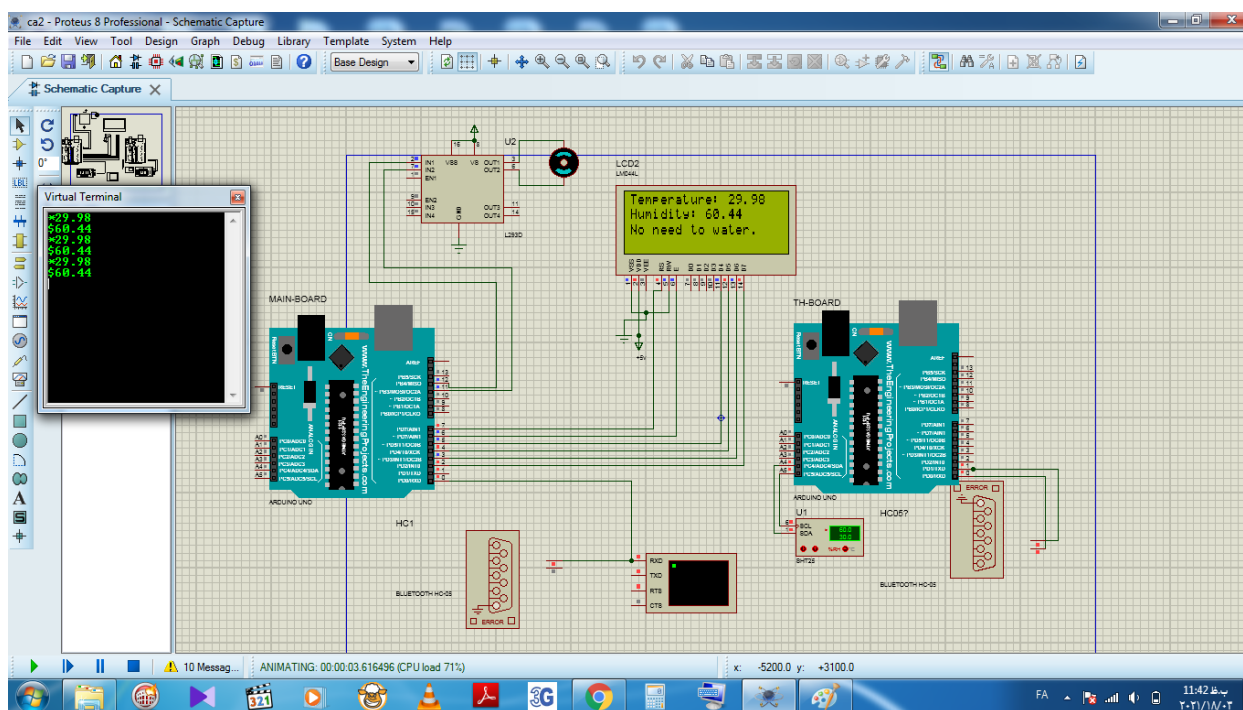
شمای کلی مدار در Proteus

دو برد Main و TH از نوع Arduino UNO انتخاب شده و خروجی کامپایل کدهایی که در قسمت قبل شرح داده شد به عنوان Program File این دو برد قرار داده شده است. موتور DC مانند پروژه قبلی به Main Board وصل شده است (خروجی های 12 و 11 برد Main به ورودی های دراپر وصل شده تا به عنوان ورودی دو سر موتور استفاده شوند). LCD از نوع LM044L انتخاب شده تا ابعاد آن 20 در 4 باشد و بین های آن مطابق با کد main board و مشخصات این LCD وصل شده است. اطلاعات سنسور دریافت می شود و بسته به شرایط پیام مناسب روی LCD چاپ می شود.

در شبیه سازی چند حالت می تواند پیش آید:

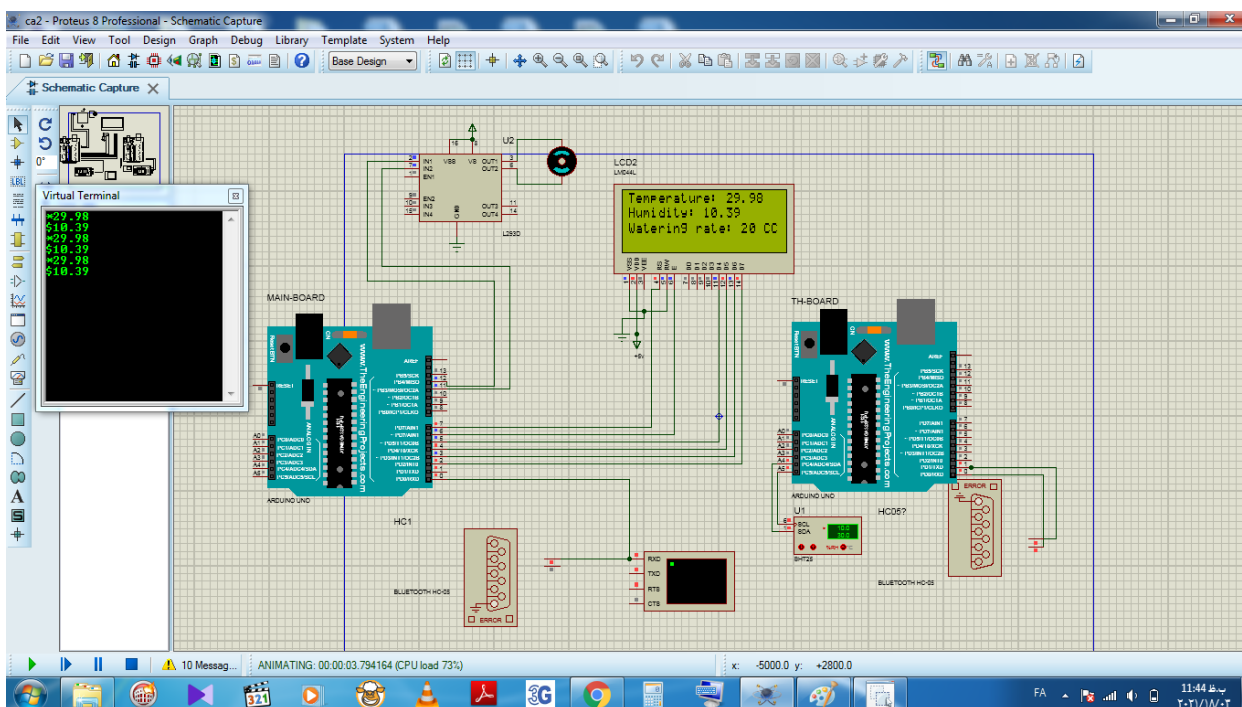
1. رطوبت بیشتر از 50 باشد

همانطور که مشاهده می‌شود، در این حالت آبیاری صورت نمی‌گیرد و روی صفحه LCD اطلاعات سنسورها مشاهده می‌شود و مشخص شده که نیازی به آبیاری نیست.



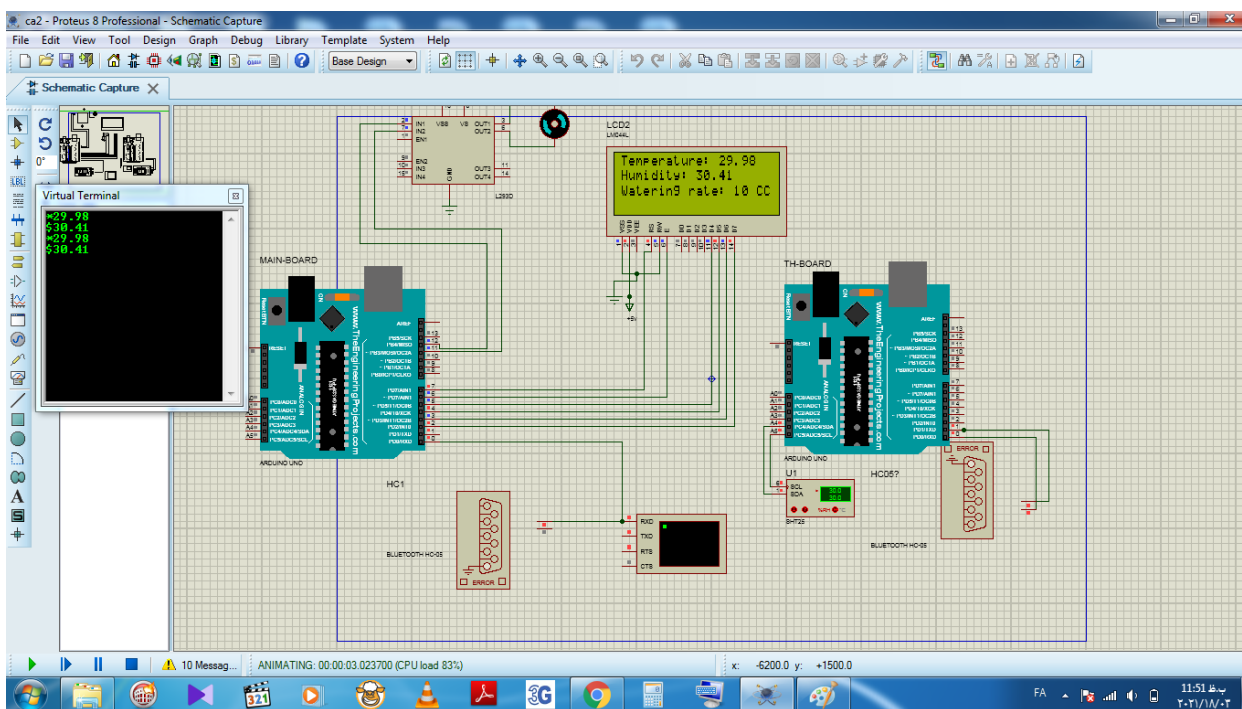
2. رطوبت کمتر از 20 باشد

همانطور که مشاهده می‌شود، در این حالت آبیاری با نرخ 20 سی سی صورت می‌گیرد و روی صفحه LCD اطلاعات سنسورها مشاهده می‌شود و مشخص شده که نرخ آبیاری 20 است و duty cycle موتور نیز 25 درصد و موتور در حال چرخش است.



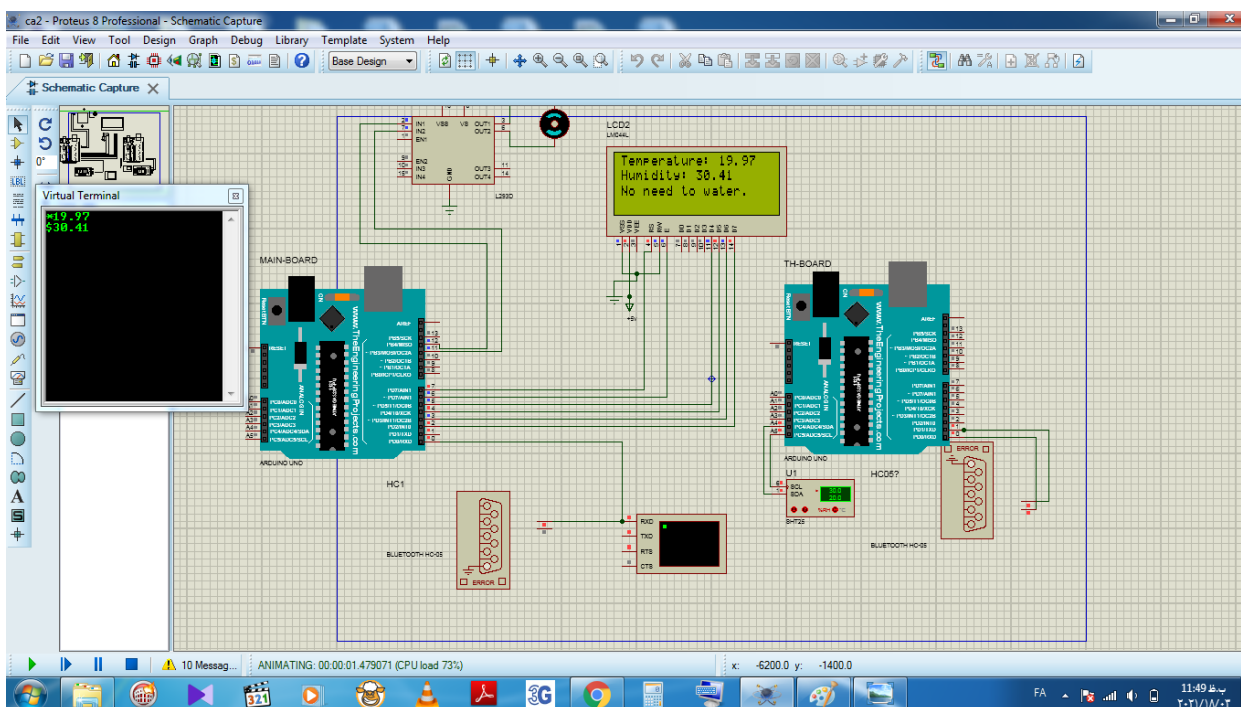
3. رطوبت بین 20 تا 50 و دما بیشتر مساوی 25 باشد

همانطور که مشاهده می‌شود، در این حالت آبیاری با نرخ 10 سی سی صورت می‌گیرد و روی صفحه LCD اطلاعات سنسورها مشاهده می‌شود و مشخص شده که نرخ آبیاری 10 است و duty cycle موتور نیز 10 درصد و موتور در حال چرخش است.



4. رطوبت بین 20 تا 50 و دما کمتر از 25 باشد

همانطور که مشاهده می‌شود، در این حالت آبیاری صورت نمی‌گیرد و روی صفحه LCD اطلاعات سنسورها مشاهده می‌شود و مشخص شده که نیازی به آبیاری نیست.



## پاسخ سوال 1

### فرکانس مورد استفاده برای ارتباط بی سیم در بلوتوث

در حالت کلی، ارتباط بی سیم در این دستگاه با استفاده از فرکانس های رادیویی صورت می پذیرد. هر دستگاه بلوتوث در باند فرکانسی 2.4 گیگاهرتز تا 4.8 گیگاهرتز کار می کند. دلیل انتخاب چنین فرکانسی این است که در سراسر جهان به طور رایگان در دسترس است و محدودیت های داشتن مجوز را ندارد. این باند فرکانس طبق یک توافق نامه بین المللی برای استفاده توسط لوازم علمی، پزشکی و صنعتی در نظر گرفته شده و ISM نامیده می شود. باند ISM به 79 کانال با پهنای باند 1 مگاهرتز تقسیم می شود. بلوتوث از لحاظ نظری دارای پهنای باند 1 مگابایت در ثانیه بوده که سرعت آن در حدود 723 کیلوبیت در ثانیه است که سرعتی مناسب برای دسترسی به اینترنت و انتقال داده ها بین وسایل محسوب می شود.



## راهکار برای رفع تداخل داده‌های ارسالی در صورت وجود چند دستگاه بلوتوث در اطراف هم

یکی از راهکارهایی که برای رفع این مشکل به نظر می‌رسد، این است که سیگنال‌های بسیار ضعیفی در حدود 1 میلی‌وات ارسال شود. فلسفه این راهکار به این برمی‌گردد که استفاده از امواج ضعیف، شعاع برد سیگنال‌های بلوتوث را به حدود 10 متر محدود می‌کند. اما ایرادی به این راهکار وارد است و آن ایراد این است که استفاده از چنین سیگنال‌های ضعیفی، خود به ایجاد تداخل بین امواج بلوتوث با امواج تلفن همراه، کامپیوتر یا دستگاه تلویزیون منجر می‌شود.

راهکار دیگری که وجود دارد و در بلوتوث از آن استفاده می‌شود، «جهش فرکانس در طیف گسترده» نامیده می‌شود. این روش احتمال استفاده از فرکانس برابر توسط دو دستگاه به صورت هم‌زمان را تقریباً به صفر می‌رساند. بر پایه این روش، به هر وسیله این امکان داده شده که از 79 فرکانس منحصر به فرد که به صورت تصادفی از میان یک سری فرکانس‌های از پیش تعیین شده انتخاب می‌شوند، استفاده کند؛ و به‌طور منظم تغییر فرکانس می‌دهد. عمل تغییر فرکانس در دستگاه فرستنده، حدود 1600 بار در ثانیه رخ می‌دهد، بر پایه این راهکار از اختلال بین امواج بلوتوث با دستگاه‌هایی مثل کنترل در پارکینگ یا تلفن‌های بی‌سیم جلوگیری می‌شود.

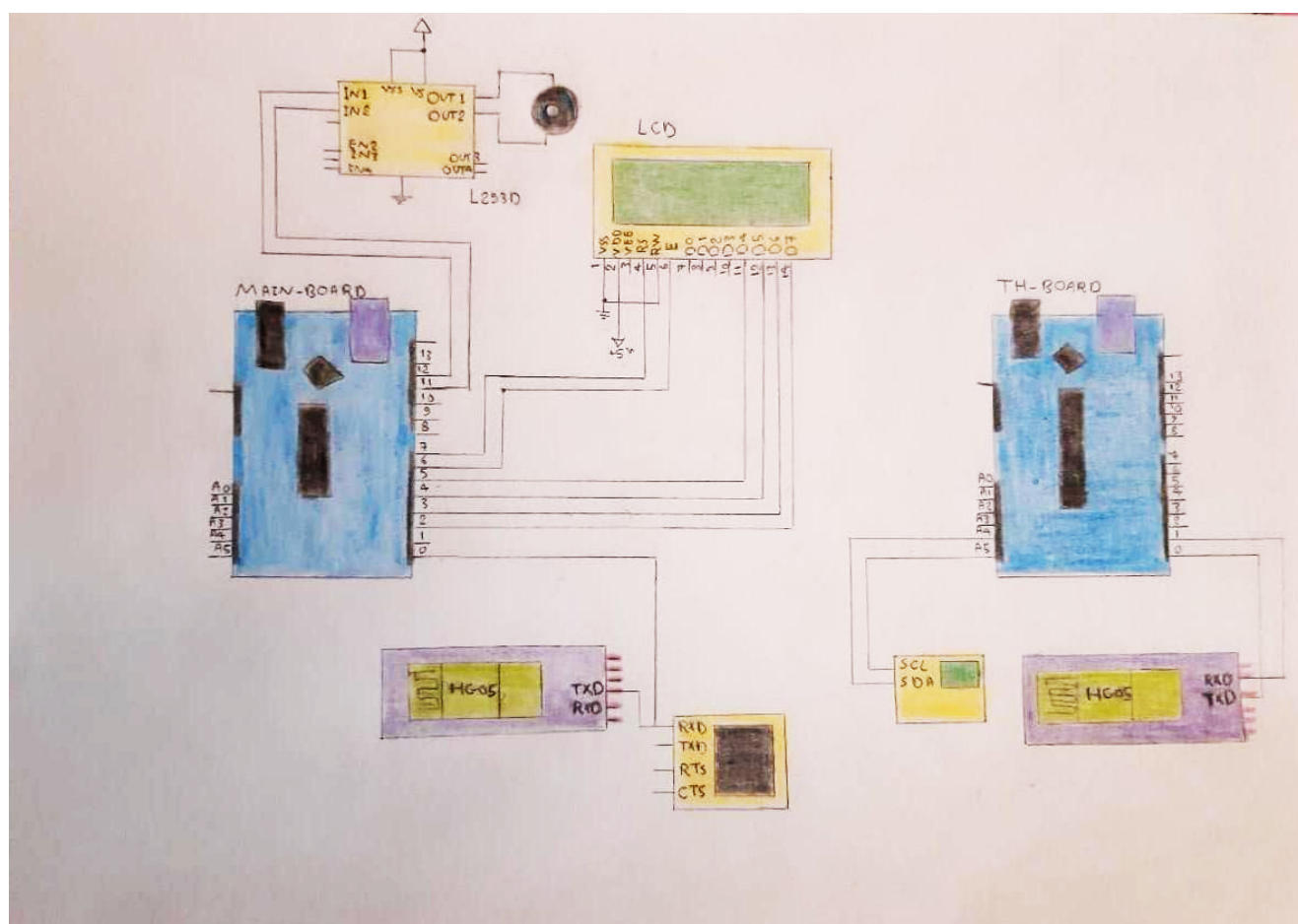
## پاسخ سوال 2

بله مطابق اسلایدهای درس، تا ۱۱۹ دستگاه را می‌توان به یک bus متصل کرد. به طور معمول، SDA و SCL یک می‌باشند و تنها در صورتی که دستگاهی بخواهد خط را ۰ کند، آن را به زمین وصل می‌کند و خط صفر می‌شود. با توجه به اینکه وجود دو مقاومت، ۱ ضعیف در خط ایجاد می‌کند، با وصل کردن به زمین و ایجاد صفر قوی، قطعاً صفر در خط قرار می‌گیرد و مشکلی ازین بابت رخ نمی‌دهد و خط هرگز نمی‌سوزد اما ممکن است داده‌های آنها با یکدیگر قاطی شوند. برای حل این مشکل باید مدلی از داوری را استفاده کنیم. دقت شود که در صورتی که

دستگاهی زودتر داده را قرار دهد، اولویت با آن است و در اینجا فرض می‌شود دو یا چند master همزمان داده‌ای را قرار می‌دهند. داوری، بیت به بیت پیش می‌رود. در هر لحظه که SCL یک می‌باشد، هر master بررسی می‌کند که وضعیت SDA با بیتی که فرستاده است برابر است یا خیر. در صورتی که هر master ای، بیت ارسالی ای مخالف وضعیت SDA داشته‌باشد، داوری را می‌بازد و تا زمان آزاد شدن خط، باید صبر کند. با توجه به توضیحات قسمت اول، در واقع هر master ای که بیت‌های صفر بیشتری را در ابتدا (MSB) ارسال کند، داوری را می‌برد چرا که صفر همواره بر یک ضعیف غالب می‌باشد. در این روش هیچ داده‌ای از دست نمی‌رود و slave هیچ دخالتی در داوری ندارد.

### پاسخ سوال 3

طراحی مفهومی مدار به شرح زیر است که در آن تلاش شده است که اتصالات و کامپوننت‌های مختلف به نمایش گذاشته شود.



تصویر طراحی مفهومی مدل مدار